CN 9 - 5
(1977. 1. 24)

ミニコンピュータ複合体のオペレーティング・システム

酒元登志志[†]　松浦敏雄[†]　前村義明[†]　矢野秀一郎[††]
藤井護[†††]　都倉信樹[†]　岡本卓爾[††††]
† 大阪大学基礎工学部　†† 富士通　††† 大阪大学大型計算機センター　†††† 岡山大学工学部

1. まえがき

ミニコンピュータ複合体については，これまでいくつかの試みがなされている．これらは，大型機なみの性能を持たせることを目的としたものと，各種の資源を共有して各ミニコン・システムの持つ長所を相互に享受し，かつその弱点を相互に補完することを目的としたものに大別できよう．[1),2)] 筆者らは先に後者の目的で既製のインタフェイス装置を用いたミニコン2台よりなる複合体のオペレーティング・システムを開発した[3)]．今回，これとは全く独立に，実験研究用のシステムとして，より柔軟で拡張性に富んだリソース・シェアリング・システムを，結合方式やインタフェイス装置の設計，製作を含めて開発した．本システムは，機器構成上の相異が比較的顕著な既設の PDP-11 3セットを対象としたもので，各システムはマイクロプロセッサ* を内蔵して融通性に富んだインタフェイス装置を介して共通バスに結合されており，ミニコンの増設にも容易に応じられる等，拡張性の高いものである．

本稿では，この複合体のリソースを共有するために開発したOSについて述べる．これは，使い勝手が良く利用者の多い単一ユーザ用の既存のOS（DOS）を改造したもので，外部仕様としてはDOSを完全に含み，他のミニコンの持つリソースをあたかも自システムが持っているかのように使用できるという意味で使い易いリソース・シェアリングの機能をもったものである．

2. 設計方針

複合体のソフトウェアは，ミニコンのそれと，マイクロプロセッサのそれとに大別される．ミニコンのソフトウェアはメーカー提供のDOS[8),9)]を基盤として下記の1)～5)の設計方針で，マイクロプロセッサのソフトウェアは6)～8)の設計方針で作製した．

1) 複合体全体の管理は，集中的には行わず，それぞれのサブシステム* のOSが行うという分散制御の方式により実現すること．
2) 各サブシステムにいるユーザ各々が，CPU，主記憶，コンソールを除く全システムのリソースを従来のDOSと全く同じ手続きで使用できること．
3) DOSの下で動作する各種のユーティリティ・プログラムが修正しなくとも実行できること．
4) DOSの内部仕様の変更を極力少なくすること．
5) 新しくサブシステムが追加されてもソフトウェアは一切変更しなくともよいこと．

---

* マイクロプロセッサとしては，設計当初に望み得た最良の INTEL 8080 を用いている．

* CPU にユニバス上のインタフェイス装置以外のすべてのデバイスを含めてサブシステムと呼ぶ

6) ミニコン間の通信に於いて，それらのオーバーヘッドがなるべく小さくてすむようなマイクロプロセッサの通信機能を実現すること．

7) 通信路が動的に開設，閉鎖でき，各サブシステム対に於いて，複数個の独立な通信路を設けることができること．

8) RAMの節約のため，リエントラント・ルーチンで構成すること．

上記の他に，ラインプリンタ等の低速デバイスに対しては，スプーリング処理を行い複数のユーザが見かけ上同時に使用できるようにすること等があげられる．

3. ハードウェア構成とその機能

図1にシステム全体の構成を示す．各ミニコン（PDP-11）[4] M_i* (i=1,2,3) のユニバスUBiは，インタフェイス装置I_iを介して共通バスCBに接続されている．

I_iはマイクロプロセッサ（INTEL 8080[5]）μ_iを内蔵している．I_iはM_iから見たとき，UBi上の単なる一つのデバイスとなっている．I_iはUBi上の任意のアドレス（M_iの主記憶の各セルやデバイスのレジスタ類）に直接アクセスでき，CB上の任意のアドレスもヌ，アクセスできる．

CB上には，RAMを用いた共有メモリCMと，システム全体の保守やデバッグのための共通コンソールCCが置かれているが，この他に任意の装置を接続することもできる．CBの制御は，UBの制御とは異なり，アービタと呼ばれる機構で簡単かつ高速に分散制御している[6]．CBはこの機構を除いてUBと全く同じ構造である．これにより，I_iからみたUBiとCBは基本的に同じとなり，I_iのハードウェア，ソフトウェア共に簡単化を図ることができる．

CMには通信のためのμのプログラムやデータが格納される．このメモリには競合防止のための1ビットのハードウェア・セマフォが設けられている．

I_iには図2では省略してあるが，単体のデバッグやモニタリングに必要

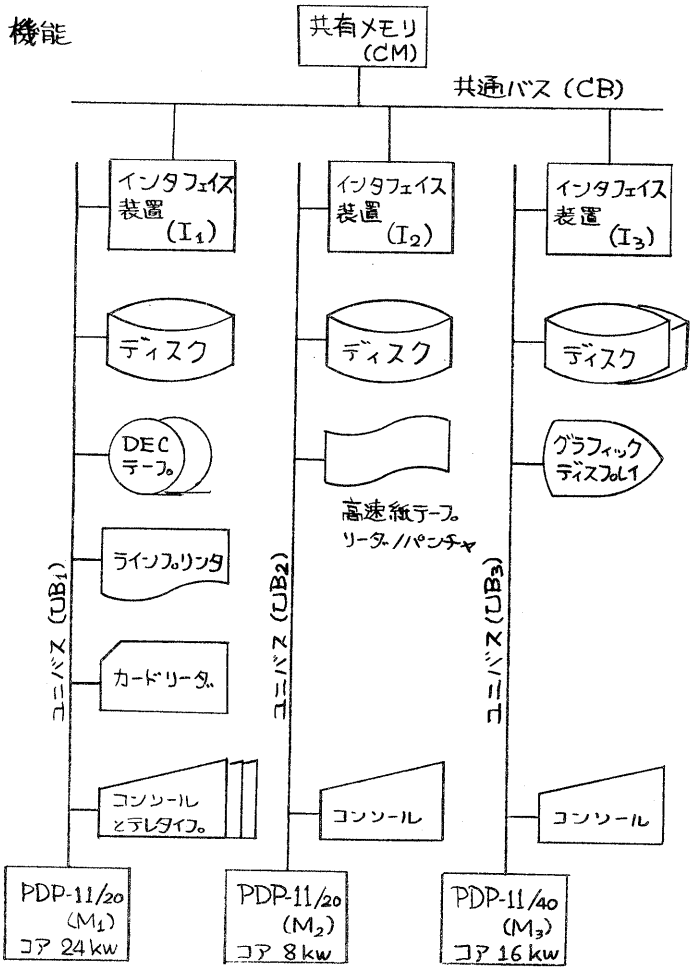

図1 システム構成

＊特に必要がなければ添字を省く

な機能を備えたコンソールμC_iがある．

以下，インタフェイス装置内の通信のために重要な役割をはたす二つの機構について述べる．

アドレス変換機構：μ_iから出力される16ビットのアドレスは，アドレス変換機構ACによってUB$_i$またはCB上のアドレスに変換される．UB$_i$及びCB上のアドレス空間は4K語（2^{13}バイト）単位のページに分割されている．ACは8コの4ビット・レジスタを内蔵しており，μ_iから出力された16ビットのアドレスは，その上位3ビットがそれで指定したアドレス変換レジスタの内容4ビットで置き換えられ，17ビットに変換される．この17ビットの最上位ビットは，残りの16ビットがUB$_i$上のアドレスかCB上のアドレスかを示す．

アドレス変換レジスタの内容は，μ_iが入出力命令によって読み書きできるほか，特定のレジスタはイニシャル・ロードのため，μC_iから手動で書き込めるようになっている．

割込み処理機構：

① M_iからμ_iへの割込み

M_iがI_i内のレジスタUIR$_i$にデータ（16ビット）を書込むとμ_iへの割込みが生ずる．このとき，前回の割込みがμ_iによって受け付けられていない場合には，UIR$_i$への二重書込みとなるので，書込む前にUIR$_i$が空かどうかを示すステータス・ビットをソフトウェアで確めねばならない．UIR$_i$にデータが入るとステータス・ビットはセットされ，CB側からの割込みとの競合を先着順で処理する割込み制御機構ICを経てμ_iに割込み要求が発せられる．μ_iは割込みを受け付けると，特定のアドレスから始まる割込み処理ルーチンを実行し，UIR$_i$の内容を読込み（このとき，ステータス・ビットは自動的にリセットされる）それを解読して所定の処理を行えばよい．

② μ_jからμ_iへの割込み

基本的には①の場合と同様である．異なるのは，複数個のマイクロプロセッサからμ_iへの割込み要求が同時的に発生する点である．これらの要求はすべて，同一のレジスタCIR$_i$にデータを書き込むことによって満たされるが，①の場合のように，CIR$_i$が空かどうかを各μ_iがソフトウェアでテストするとすれば，その間，μ_jがCBの使用権を確保しなければならず，そのためのハードウェアが必要となり，時間のロスも大きい．そこで本システムでは，CB

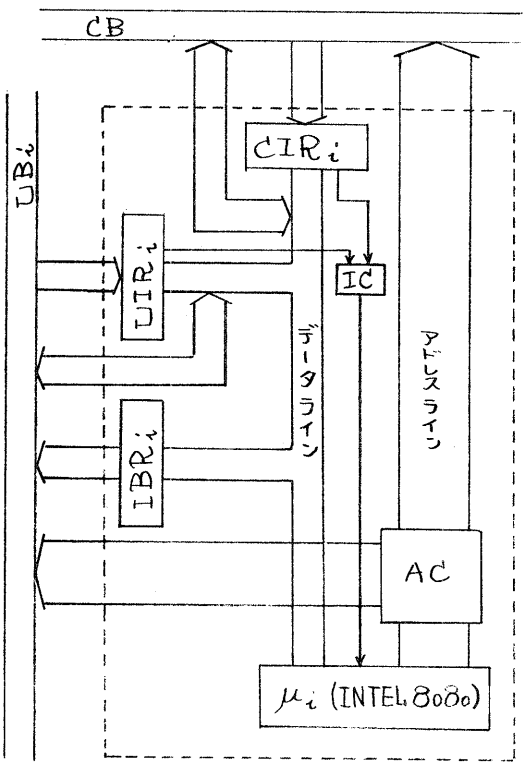

図2　インタフェイス装置

からのCIRiの使用は，ハードウェアで実現した“テスト・アンド・セット”の機構を用いている．

③ μiからMiへの割込み

Miの割込み機構をそのまま用いている．μiが出力命令でレジスタIBRiにデータ（16ビット）を書込むと，Miに割込み要求が発せられ，Miの割込み処理ルーチンが実行される．μiがIBRiにデータを二重書込みしないように，ここにも“テスト・アンド・セット”機構を用いている．

4．ソフトウエア

4.1 I/O要求処理

DOSはユーザがコンソールを通じてファイル管理プログラム，コンパイラ，実行制御ルーチンなどと通信を行いながら，インタラクティブにプログラムの作製，実行を行うのに適したOSである．しかし，単一ユーザ用OSであるためマルチプログラミングの機能は持っていない．

4.1.1 従来のI/O要求処理

ユーザは表1に示すようなI/Oマクロを発することによりデバイスを利用することができる．デバイスは表2a欄に示す2文字のデバイス名によって指定される．

ユーザの発したI/OマクロはI/Oマクロ解析ルーチン（EMTH：EMTハンドラ）で解析され，各々のロジカルI/Oルーチン（LIO）で処理され，さらに必要ならば，フィジカルI/Oルーチン（PIO）の処理を受ける（図3参照）．LIOは一般に主記憶上に常駐でなく，EMTHによって主記憶上の特定の領域（MSB：モニタ・スワップ・バッファ）にロードされ，ファイルの登録や管理，オープン中のファイルをロックする作業，二次記憶装置に於ける記憶領域の使用状態を示すビットマップの管理等を行う．PIOは各デバイスに対して用意されており，LIOからの要求により，デバイスの制御装置を直接操作し，データの転送，デバイスからの割込みの処理を行う．

.INIT	目的のデバイスに対するPIOをロードする．
.OPEN	ファイルを“開く”
.READ	ファイルから1レコード読込む
.WRITE	ファイルに　1レコード書込む
.CLOSE	ファイルを“閉じる”
.RLSE	目的のデバイスに対するPIOをはなす
.RENAME	ファイルの名前をかえる
など	

表1　I/Oマクロとその処理

デバイス	a	b
ディスク	DK	DKx
デックテープ	DT	DTx
ラインプリンタ	LP	LPx
カードリーダ	CR	CRx
キーボード	KB	——
など		

xは1文字のサブシステム名

表2　デバイス名

4.1.2 複合体でのI/O要求処理の方式

上述のようなDOSのI/O処理機能を基盤としてリソース・シェアリングを行うには，これらのI/Oマクロをいかに処理するかが問題となる．これには次の2つの方法が考えられる*．

---

* このほか，ユーザが他のサブシステムにあるファイルを使うに先だってそのファイルを自サブシステムの二次記憶装置にコピーするというファイルレベルでの方法もあるが，これは二次記憶装置の使用効率が低下するだけでなく，ファイル管理プログラムの大幅な変更を必要とする．

① ＰＩＯレベルで処理を行う方法

他のサブシステムに属すデバイスに対しても自システム内にＰＩＯ（擬似ＰＩＯ）を用意し，この擬似ＰＩＯがそのデバイスを持つサブシステムのＰＩＯと通信を行って目的の処理を達する．

② ＬＩＯレベルで処理を行う方法

ユーザが他のサブシステムに属するデバイスに対するＩ／Ｏマクロを発すると，ＯＳは相手のＯＳにそれを送信する．相手のＯＳはそのＩ／Ｏマクロの処理を行い，結果を返す．

以上の方法を比較すると，①の方法では，ファイルやビットマップの管理はＬＩＯが行うので，１つのファイルや１つのデバイスのビットマップの管理を同時に多数のサブシステム内にあるＬＩＯが行うことになり，それらの間で非常に多くの通信を行う必要がある．又，デバイスが追加されるごとに擬似ＰＩＯを作製しなければならない．②の方法では，①の方法に比べ，Ｉ／Ｏ処理依頼を受けた側のＯＳのオーバーヘッドは大きいが，ファイルやビットマップの管理は単一のＬＩＯのみが行うので通信の回数は少なくてすむ．又，デバイスが新しく追加されても手を加える必要はない．以上の理由から②の方法を採用することとした．

4.1.3 外部仕様

本システムを利用するユーザは空いているコンソールを探し，従来と同じようにログイン操作を行う．以後はすべてのリソースが自分の確保したサブシステムに属しているものとみなして以下の点を除き従来のＤＯＳと全く同じ手続きでシステムを利用できる．他サブシステムに属するデバイスを使用する場合には，従来のデバイス名にサブシステム名を表わす１文字を付け加えた３文字のデバイス名（表２ｂ欄）を用いる．デバイス名はコンソールからアサイン・コマンドで簡単に指定，変更できるので，ＤＯＳの下で既に開発されたプログラムで他のサブシステムのリソースを使用することも容易である．

4.1.4 内部仕様の概要

ユーザが発したＩ／Ｏマクロは ＥＭＴＨで解析され，自サブシステム内のデバイスに対するものであれば従来と同じ処理が行われ，そうでなければ，即ち，他のサブシステムのデバイスに

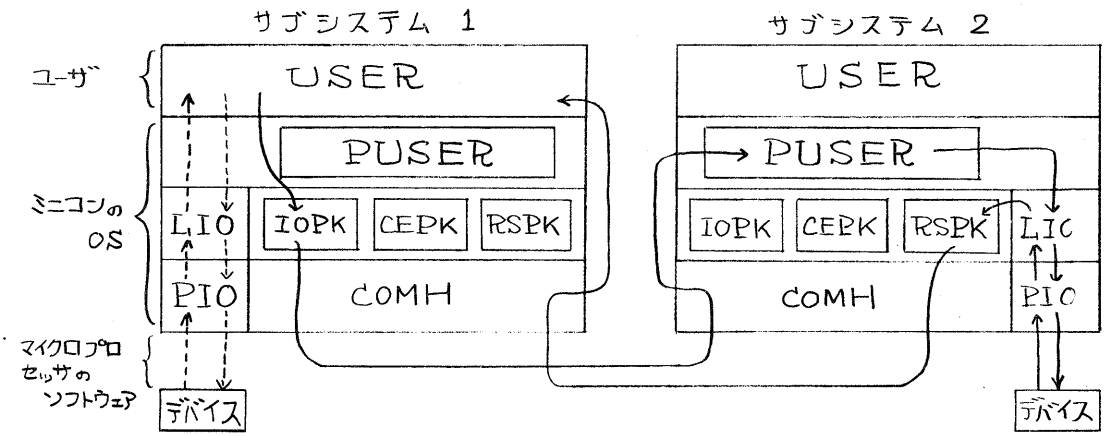

図３ Ｉ／Ｏ要求処理の流れ

対するI/O要求であれば，それはI/O要求パックルーチン（IOPK）により16ワードのメッセージ・コントロール・ブロック（MCB，図4）にパックされ通信ハンドラ（COMH）に渡される．COMHはこれをユーザが指定したデバイスをもつサブシステムに転送するためにμに割込みをかける．割込みを受けたμは転送ルーチンを実行し，MCBを相手サブシステムに送る．このMCBを受け取ったサブシステムは，擬似ユーザルーチン（PUSER）に制御を渡し，実際のI/O処理を実行し，その結果を依頼サブシステムに返す（図3参照）．

相手サブシステム番号
自サブシステム番号
LIOルーチン番号又はKB
I/O制御ブロック 又は キーボード・イメージ

図4　MCB

4.2　マイクロプロセッサのソフトウェア

2つのプロセスP，Q間の情報授受は，あらかじめ開設されたルートを介して行われる．ルートは全二重の通信路に対応するもので，P，Qは同じ口ジカルなルート名を指定してμに要求することによってフィジカルなルート番号を与えられ，ルートの開設が行われる．

M_i－μi 間，μi－μj 間の通信はすべて16ビットデータを伴う割込みによって行われる．以後，この16ビットデータを割込みパラメータと呼ぶ．この割込みパラメータには，図5に示す3つのモードがある．

Aモードは，μに1バイト以下のデータで完結する仕事を依頼するためのものである．

Bモードは，ミニコン間で1バイトのメッセージの授受を容易に行うためのものである．割込みパラメータのうち8ビットがメッセージのために使用されるので，残りの6ビットで送り手と受け手を指定しなければならないが，ルート方式によると64組（最大128プロセス）の通信が可能である（両者をあらわに指定する方式では，最大でも8プロセスしか通信に参加できない）．

Cモードは，1バイトを越えるデータをμに与えて仕事を依頼するのに用いられる．

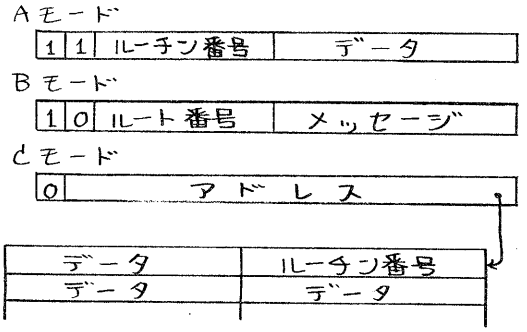

図5　割込みパラメータのモード

ルートの開設

互いに通信を希望する2つのプロセスP_1，P_2（それぞれM_1，M_2上にある）は，ロジカルルート名等の通信に必要な情報を書いたテーブル（Uテーブルと呼ぶ，図6-(a)参照）を準備し，このテーブルアドレスを記したCモードのパラメータを送ってμに割込むことによってルート開設要求を出す．この要求はP_1とP_2で非同期に発せられるが，以下，P_1が先にルートの開設要求を出したものとしてμの動作を述べる．

先ず，P_1からの割込みによってμ_1はルート登録ルーチンを実行し，相手のプロセスのルート開設要求がすでに登録済みであるかどうかを調べる．この場合まだ登録されていないので，C

M内にロジカルルート名でマッチングをとるためのランデブーテーブル[7]*の要素（Cテーブルと呼ぶ）を作り，登録を完了する．次に，P_2が要求を出したとき，今度はP_1が登録したCテーブルがすでにできているので，このCテーブル上でマッチングがとれる．続いてルート登録ルーチンは使われていないルート番号を，Cテーブル及びP_1とP_2のUテーブルに書込み，ルートの開設を完了する．

Uテーブル・リンク・ポインタ
*1 ルート登録ルーチンのルーチン番号
*1 パラメータ
*2 ルート番号
*2 ルートステータス
*1 ロジカルルート名
*1 ユーザコード
*1 通信相手のユーザコード

(a) ルート開設時

Uテーブル・リンク・ポインタ
*3,*4 ブロック転送ルーチンのルーチン番号
パラメータ
ルートステータス
*3 送り手側バッファポインタ
*3 送り手側データサイズ
*4 受け手側バッファポインタ
*4 受け手側バッファサイズ
転送された データ数

(b) ブロック転送時

*1---ルート開設前にM_iがセットすべきところ
*2---ルート開設時にμ_iがセットするところ
*3---送り手側がセットすべきところ
*4---受け手側がセットすべきところ

図6 Uテーブル

ルートの閉鎖

P_2と通信を行っていたP_1は，ルート閉鎖ルーチンのルーチン番号とルート番号よりなるAモードのパラメータを送ってμ_1に割込みをかけることによりルートを閉鎖できる．

この割込みにより，μ_1はルート閉鎖ルーチンを実行して，μ_2を介してP_2のUテーブル内のルートが閉設されていることを示すビットをたおす．さらにCM内の対応するCテーブルを消去し，使用していたルート番号を解放する．以上でルートの閉鎖が完了する．

バイト転送

P_1，P_2間にルートが開設されているとき，P_1からP_2に1バイトのデータを転送するのは次のように行われる．

P_1はルート番号と転送したい1バイトのデータを1ワードにパックしたBモードのパラメータを送ってμ_1に割込みをかける．

この割込みを受けたμ_1はパラメータ中のルート番号とCテーブルから割込むべき相手を知り，μ_2に割込みをかける．μ_2は割込まれたときと同じパラメータを用いてM_2に割込みをかける．このようにして，プロセスP_1からP_2へのバイト転送が実現される*．

ブロック転送

P_1，P_2間にルートが開設されているとき，P_1からP_2にnバイトのデータを転送するのは次のように行われる．

P_1は図6-(b)で示すようにUテーブルをセットし，Uテーブルのアドレスを含むCモードのパラメータを送ってμ_1に割込みをかける．

この割込みによりμ_1はUテーブル内のルーチン番号で示されるブロック転送ルーチンを起動し，CM内にバッファを確保して，転送すべきデータをコピーし，μ_2に割込んでブロック転

*ランデブーテーブルはリスト構造になっている．

*転送が終って，データを受け取ったことを送り手側のプロセスに知らせるには返答ルーチン番号とルート番号よりなるAモードのパラメータを用いてμに割込めばよい．これによって，送り手側のUテーブル内の転送終了を示すステータス・ビットがセットされる．

送のルーチンを起動する．このルーチンは，P_2のUテーブルをみて，大きさがnバイト以上のバッファが用意されているかを調べる．バッファが用意されていないとき，あるいはバッファの大きさが小さいときは，転送したいデータの大きさをUテーブルに書き込み，M_2に割込む．すでに十分な大きさのバッファが用意されているときは，そのバッファにデータを書き，さらに転送したデータの大きさをUテーブルに書込んでM_2に割込む．前者の状態で割込まれた場合，P_2が十分な大きさのバッファを用意してμ_2に割込めば，μ_2はもう一度同じルーチンを起動してデータを転送する*．

4.3. ミニコンのソフトウェア

4.3.1 修正されたソフトウェア

リソース・シェアリングを実現するOSを作製するにあたり，従来のDOSを構成するモジュールのいくつかに修正が加えられたが，以下に，4.3.2に関連のある機能を追加されたものについて述べる．

イニシャライズ・ルーチン

① 新たに作製されたモジュールのためのワーキング・エリア（Uテーブル，MCBのリンク・スタートやサブシステムのアイデンティファイヤ等）の確保と初期設定をする機能が追加された．

② 下記の用件に関しOS間で通信を行う必要がある．

1) I/Oマクロの処理を依頼する．
2) I/Oマクロの処理の結果を返す．
3) 依頼主にエラー・メッセージを伝える．
4) 依頼したI/Oマクロ処理に関するコンソールからの指示（中止や再開など）を伝える．
5) 1)～4)に対する返事を伝える．

これらの通信はOS間に開設されたルートを用いて行われる．そこで，各M_iのシステム・イニシャライズ時にM_j $(j \neq i)$のOSとのルートを開設するようにμ_iに要求を出す機能が追加された．

このような要求を出したOSのすべての組合せに対してルートが開設される．M_iのOSはM_jのOSとのルートが開設されれば，M_jのリソースを共有できると判断する．

③ 従来のDOSはシステム・イニシャライズ時に自システムに属するデバイスについて，そのPIOのサイズや開始番地等が記入されたリスト（DDL）を完成する．このDDL（図7-(a)参照）だけでは他のサブシステムに属するデバイスに関する情報は与えられないので，複合体のOSにはそれらの情報（そのデバイスが属するサブシステム名等）をもったDDLの要素（図7-(b)参照）を生成する機能が付け加えられた．

2文字のデバイス名（表2 a欄参照）
開始番地
割込みベクトルアドレス
二次記憶内の格納場所

(a) 従来のDDLの要素

3文字のデバイス名（表2 b欄参照）
未使用
デバイスが属するサブシステム名
2文字のデバイス名

(b) 追加されたDDLの要素

図7 DDLの構成要素

EMTH

他のサブシステムに属するデバイスに対するI/Oマクロは自サブシステム内では処理できず，指定されたデバイスをもつサブシステムに依頼する必要がある．このために，EMTHに以下の機能が付け加えられた．

---

*前頁右下の脚注に同じ．

DDLを調べ，I/Oマクロで指定されたデバイスが他のサブシステムに属するなら，それを管理するOSに対し，I/Oマクロ処理の依頼をするためにIOPKをコールする．

4.3.2 追加されたソフトウェア

以下にリソース・シェアリングを実現するOSのために新たに作製されたルーチンとその機能を示す．

PUSER

他のサブシステムから送られてきたMCBを解析し，依頼されたマクロを自システムのEMTHに発するルーチンである

PUSERは他サブシステムからの割込みによって起動されるため，直ちにMCBを調べてI/Oマクロを発した場合，他のLIOが処理中であると下に述べるようにデッドロックが生じる．そこでPUSERはMCBを一旦待ち行列につなぎ，他のLIOの処理が終ってからMCBの処理を始める．

デッドロックの生じる理由：DOSは単一ユーザ用OSであるため，一時に一つのI/Oマクロを処理するようにしか設計されていないので，MSBの使用に関してデッドロックが起こりうる．即ち，MSB中にあるLIOが実行中であるときに，割込みによりPUSERに制御が移ると，このLIOは未終了でありMSBを専有し続ける．PUSERがI/Oマクロをその状態で発すると，MSBが使えずI/Oマクロの処理は待ち状態に入り，進行しない．MSBを2つ設ける方法も考えられるが，これは，モニタ領域が大きくなるばかりでなく，LIOの大幅な改造を必要とする．

IOPK

他サブシステムへのI/O要求に必要な各種のパラメータを16ワードのMCB（図4参照）にまとめて，COMHに渡す（この中にはI/O要求の詳細なパラメータが記されたユーザが用意したI/O制御ブロックが含まれている．）．データの転送を必要とするI/O要求（例えば .WRITE）では，MCBに先だち，DTB（データ転送ブロック：相手サブシステム番号，自サブシステム番号，データバッファへのポインタ等からなる）を作成し，COMHに渡す．

RSPK

他サブシステムからのI/O要求が処理されたとき，依頼したサブシステムへその処理結果を返すルーチンである．そのためのDTBをつくり，COMHに渡す．

CEPK

依頼されたI/O要求を処理中に発生したエラーに関するメッセージは依頼したサブシステムのコンソールに出力した方がよい．また，依頼したI/O要求の中止，再開等を指示するコンソール・コマンドは相手のサブシステムに伝える必要がある．このルーチンはこれらのメッセージやコマンドを相手のサブシステムに伝えるためのものであり，MCBをつくり，COMHに渡す．

COMH

これはOSルートを使ってデータの授受を直接行うルーチンである．

MCBの処理：M_1 から M_2 に転送する場合を例として述べる．この場合，IOPK又はCEPKから渡されたMCBには相手サブシステム名として［M_2］が書かれているので，M_2 との間に開設されたOSルートの転送待ち行列にそれをつなぐ．このとき，待ち行列が空なら直ちに μ_1 にブロック転送の要求を出す．M_2 からMCB受け取りの通知を受けた M_1 のCOMHは M_1-M_2 間のOSルートの転送待ち行列を調べ，空でなければ，再び μ_1 にブロック転送の要求を出す．

MCBを受け取った M_2 のCOMH

は，そのことをバイト転送でM1にしらせる。
DTBの処理：MCBの場合とほぼ同じであるが，DTB内のバッファポインタで示されるデータを転送する点が異なる。

4.4 エラー処理

他のサブシステムから依頼されたI/O要求の処理中に起こりうるエラーの種類は大きく分けて2つである。それぞれのエラーの特徴及び処理について述べる。

フェイタル・エラー：これは必要な領域が確保できなかったとか，システムにないデバイスへ要求を出したときなどに生じ，回復は不可能である。この種のエラーが生じたときは，依頼主にエラー・メッセージを送り，待ち行列内の次の要求を処理する。エラーが生じた要求は放棄される。

アクション・エラー：使用するデバイスがオフラインになっていたときなどに生じ，ある動作（デバイスをオンラインにする等）を行った後，処理の続行が可能なものである。この種のエラーが生じたときには，依頼主にエラー・メッセージを送り，依頼主から処理再開の要求が来れば続行できるようにエラー発生時の状態を記憶しておいて，次の要求の処理に移る。

サブシステム間のブロック転送に関しては，受け手側のμPがチェックサムでエラーの検出を行い，エラーを検出したときは送り手側のMに知らせる。

5. むすび

複合体のソフトウェアの開発に際して苦心した点は，DOSが単一ユーザ用OSであるのでその制約をいかに吸収するかということ，DOSのモニタ変更時のデバッグ，マイクロプロセッサの少ないレジスタでのリエントラントルーチンの作製等があげられる。

マイクロプロセッサのOSのサイズは3Kバイトであり，ミニコンのOSは2Kバイト増加し49Kバイトとなり，常駐部は1.0Kバイト増加し4.7Kバイトとなった。また，他サブシステムにもタスクを発生させる機能を設計製中である。これにより，システムの有用性が更に向上すると思われる。

日頃御指導頂く大阪大学嵩忠雄教授，熱心に御討論頂いた嵩研究室の諸氏および御協力頂いた岡山大学電子回路研究室の諸氏に深く感謝します。とくに，谷口健一，故細見輝政，奥井順，葛山善基，本藤勉の諸氏に厚く感謝します。

―参考文献―

1) 小特集：コンピュータ・コンプレックス，情報処理，Vol. 15, No. 7, pp. 524～546 (1974)
2) 徳田他：KOCOSのアーキテクチャ，情報処理学会計算機アーキテクチャ研究会資料 (1975)
3) 本田他：PDP-11/20 デュアルシステムにおけるデバイスシェアリングシステムの製作，情報処理，Vol. 14, No. 10, pp. 794～801 (1973)
4) PDP-11 Processor Handbook, Digital Equipment Corp. (1971)
5) Programming Manual for the 8080 Microcomputer System, パネトロン株式会社 (1974)
6) 岡本他：非同期式リングアービタの一方式，信学論D, Vol. J59-D, No. 8, pp 582～583 (1976)
7) D.C. Walden : A System for Inter-process Communication in a Resource Sharing Computer Network, Comm. ACM, Vol. 15, No. 4 pp. 221～230 (1972)
8) PDP-11 DOS/BATCH Software Support Manual, Digital Equipment Corp. (1974)
9) PDP-11 Disk Operating System Monitor Programmer's Handbook, Digital Equipment Corp. (1972)
10) 松浦他：ミニコンピュータ複合体とそのオペレーティングシステム，情報処理採録決定